AMENDMENTS TO THE CLAIMS

Claim 1. (currently amended): A circuit having support for normalization of significants, ~~significands,~~ comprising:

a first register block, said first register block including at least one first register for holding a first exponent and a first significant ~~significand~~ of a first floating point number and a first logic capable of left shifting the significant ~~significand~~ of the first floating point number;

a second register block, said second register block including at least one second register for holding a second exponent and a second significant ~~significand~~ of a second floating point number;

a1

a plurality of flags, ~~coupled to said first register block,~~ each of said plurality of flags having a state based on the contents of said first significant; ~~significand~~;

an arithmetic logic unit coupled to said first register block, said second register block, and said plurality of flags, said arithmetic logic unit causing the first logic to left shift the first significant ~~significand~~ based upon the states of said plurality of flags.

Claim 2. (currently amended): The circuit of claim 1, wherein said plurality of flags further comprises:

an I^{th} flag, wherein I is a non-negative integer, said I^{th} flag which is set to a first sate when the 2^I most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the 2^I most significant bits is non-zero.

Claim 3. (currently amended): The circuit of claim 2, wherein said arithmetic logic unit causes said first logic to left shift by 2^I bits the first significant ~~significand~~ if said I^{th} flag is set to the first state.

Claim 4. (original): The circuit of claim 3, wherein said arithmetic logic unit is coupled to a temporary storage location for storing an adjustment to be subtracted from said first exponent, and increments said adjustment by 2^I if said first flag is set to the first state.

Claim 5. (original): The circuit of claim 2, wherein I is 0.

Claim 6. (original): The circuit of claim 2, wherein I is 1.

Claim 7. (original): The circuit of claim 2, wherein I is 2.

Claim 8. (original): The circuit of claim 2, wherein I is 3.

Claim 9. (original): The circuit of claim 1, wherein said arithmetic logic unit is coupled to a temporary storage location.

Claim 10. (original): The circuit of claim 9, wherein said temporary storage location is a register in a register file.

Claim 11. (original): The circuit of claim 9, wherein said temporary storage location is a main memory accessed through a memory interface.

a\

Claim 12. (currently amended): The circuit of claim 1, wherein:

said plurality of flags further comprises,

an I^{th} flag, wherein I is a positive integer of at least 3, which is set to a first state when the 2^I most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the 2^I most significant bits of said first significant ~~significand~~ is non-zero;

an $(I-1)^{th}$ flag which is set to a first state when the $2^{(I-1)}$ most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the $2^{(I-1)}$ most significant bits of said first significant ~~significand~~ is non-zero;

an $(I-2)^{th}$ flag which is set to a first state when the $2^{(I-2)}$ most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the $2^{(I-2)}$ most significant bits of said first significant ~~significand~~ is non-zero; and

an (I-3)th flag which is set to a first state when the $2^{(I-3)}$significant bits of said first significant ~~significand~~ are each zeros and a second state if the $2^{(I-3)}$ significant bits of said first significant ~~significand~~ is non-zero; and wherein

said arithmetic logic unit is coupled to a temporary storage location, said arithmetic logic unit initially setting the temporary storage location to zero, then modifying said temporary location based upon the state of the plurality of flags, and finally modifying said first exponent based on the contents of said temporary location.

Claim 13. (original): The circuit of claim 12, wherein said temporary storage location is a register in a register file.

a1

Claim 14. (original): The circuit of claim 12, wherein said temporary storage location is a main memory accessed through a memory interface.

Claim 15. (original): The circuit of claim 12 wherein said arithmetic logic unit modifies the first exponent by subtracting the contents of said temporary location from said first exponent.

Claim 16. (original): The circuit of claim 12, wherein I is equal to 3.

Claim 17. (currently amended): A massively parallel processing system, comprising:

a main memory;

an array of processing elements, each processing element of the array being coupled to said main memory and other processing elements of said array, wherein each of said processing elements comprises,

a first register block, said first register block including at least one first register for holding a first exponent and a first significant ~~significand~~ of a first floating point number and a first logic capable of left shifting the significant ~~significand~~ of the first floating point number;

a second register block, said second register block including at least one second register for holding a second exponent and a second significant ~~significand~~ of a second floating point number;

a plurality of flags, ~~coupled to said first register block,~~ each of said plurality of flags having a state based on the contents of said first significant; ~~significand;~~

an arithmetic logic unit coupled to said first register block, said second register block, and said plurality of flags, said arithmetic logic unit causing the first logic to left shift the first significant ~~significand~~ based upon the states of said plurality of flags.

Claim 18. (currently amended): The massively parallel processing system of claim 17, wherein said plurality of flags further comprises:

an I^{th} flag, wherein I is a non-negative integer, said I^{th} flag which is set to a first sate when the 2^I most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the 2^I most significant bits is non-zero.

Claim 19. (currently amended): The massively parallel processing system of claim 18, wherein said arithmetic logic unit causes said first logic to left shift by 2^I bits the first significant ~~significand~~ if said I^{th} flag is set to the first state.

Claim 20. (original): The massively parallel processing system of claim 19, wherein said arithmetic logic unit is coupled to a temporary storage location for storing an adjustment to be subtracted from said first exponent, and increments said adjustment by 2^I if said first flag is set to the first state.

a1

Claim 21. (original): The massively parallel processing system of claim 18, wherein I is 0.

Claim 22. (original): The massively parallel processing system of claim 18, wherein I is 1.

Claim 23. (original): The massively parallel processing system of claim 18, wherein I is 2.

Claim 24. (original): The massively parallel processing system of claim 18, wherein I is 3.

Claim 25. (original): The massively parallel processing system of claim 17, wherein said arithmetic logic unit is coupled to a temporary storage location.

Claim 26. (original): The massively parallel processing system of claim 25, wherein said temporary storage location is a register in a register file.

a1

Claim 27. (original): The massively parallel processing system of claim 25, wherein said temporary storage location is a main memory accessed through a memory interface.

Claim 28. (currently amended): The massively parallel processing system of claim 17, wherein:

said plurality of flags further comprises,

an I^{th} flag, wherein I is a positive integer of at least 3, which is set to a first state when the 2^{I} most significant bits of said first significant ~~significand~~ are each zeros and a

second state if any of the 2^I most significant bits of said first significant ~~significand~~ is non-zero;

a $(I-1)^{th}$ flag which is set to a first state when the $2^{(I-1)}$ most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the $2^{(I-1)}$ most significant bits of said first significant ~~significand~~ is non-zero;

a $(I-2)^{th}$ flag which is set to a first state when the $2^{(I-2)}$ most significant bits of said first significant ~~significand~~ are each zeros and a second state if any of the $2^{(I-2)}$ most significant bits of said first significant ~~significand~~ is non-zero; and

a $(I-3)^{th}$ flag which is set to a first state when the $2^{(I-3)}$ most significant bits of said first significant ~~significand~~ are each zeros and a second state if the $2^{(I-3)}$ significant bits of said first significant ~~significand~~ is non-zero; and wherein

a1

said arithmetic logic unit is coupled to a temporary storage location, said arithmetic logic unit initially setting the temporary storage location to zero, then modifying said temporary location based upon the state of the plurality of flags, and finally modifying said first exponent based on the contents of said temporary location.

Claim 29. (original): The massively parallel processing system of claim 28, wherein said temporary storage location is a register in a register file.

Claim 30. (original): The massively parallel processing system of claim 28, wherein said temporary storage location is a main memory accessed through a memory interface.

Claim 31. (original): The massively parallel processing system of claim 17 wherein said arithmetic logic unit modifies the first exponent by subtracting the contents of said temporary location from said first exponent.

Claim 32. (original): The massively parallel processing system of claim 18, wherein I is equal to 3.

Claim 33. (currently amended): A method for normalizing the significant ~~significand~~ of a floating point number stored in a processing element having an exponent register, a plurality of significant ~~significand~~ registers, an I^{th} flag indicating whether the 2^I most significant bits of the significant ~~significand~~ are each zero, a $(I-1)^{th}$ flag indicating whether the 2^I most significant bits of the significant ~~significand~~ are each zero, a $(I-2)^{th}$ flag indicating whether the $2^{(I-2)}$ most significant bits of the significant ~~significand~~ are each zero, a $(I-3)^{th}$ flag indicating whether the $2^{(I-3)}$ most significant bit of the significant ~~significand~~ is zero, and a temporary variable, wherein I is an integer of at least 3, said method comprising the step of:

a1

(a) initializing the temporary variable to zero;

(b) if said I^{th} flag is set, left shifting the significant ~~significand~~ by 2^I bits and incrementing the temporary variable by 2^I;

(c) if said $(I-1)^{th}$ flag is set, left shifting the significant ~~significand~~ by $2^{(I-1)}$ bits and incrementing the temporary variable by $2^{(I-1)}$;

(d) if said (I-2)th flag is set, left shifting the significant ~~significand~~ by $2^{(I-2)}$ bits and incrementing the temporary variable by $2^{(I-2)}$;

(e) if said (I-3)th flag is set, left shifting the significant ~~significand~~ by $2^{(I+3)}$ bit and incrementing the temporary variable by $2^{(I-3)}$; and

(f) decrementing the exponent register by the value of the temporary variable.

Claim 34. (original): The method of claim 33, wherein I is equal to 3.

Claim 35. (original): The method of claim 33, wherein step (a) is performed before step (b).

a1

Claim 36. (currently amended): The method of claim 33, ~~43,~~ wherein step (c) is performed after step (b).

Claim 37. (currently amended): The method of claim 34, ~~44,~~ wherein step (d) is performed after step (c).

Claim 38. (currently amended): The method of claim 35, ~~45,~~ wherein step (e) is performed after step (d).

Claim 39. (currently amended): The method of claim 36, ~~46,~~ wherein step (f) is performed after step (e).

a'